Administering SAS Viya on Kubernetes

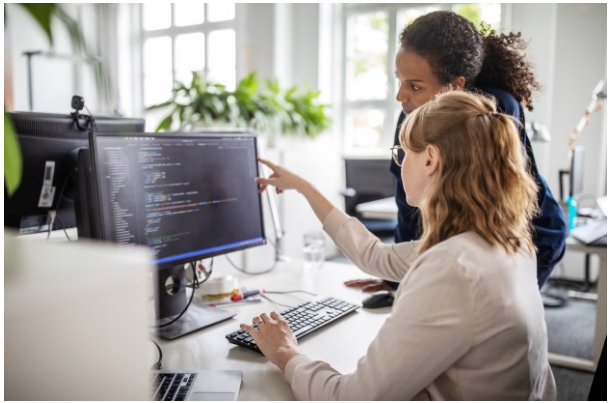
SAS Viya - Administration

FANS Oslo Platform meeting February 8. 2023

Ole-Martin Hafslund

SAS Viya Admin vs. K8S Admin

Kubernetes Administrators and SAS Viya Administrators



Essential SAS Administration Tasks Have Not Changed

Routine tasks must still be performed

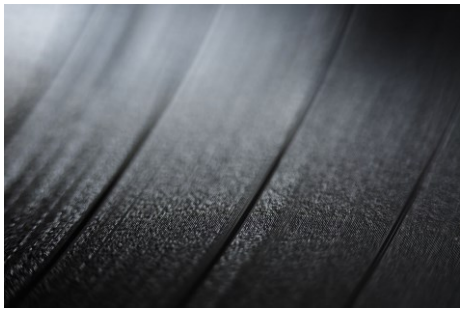
- ✓ Identity management
- ✓ Securing folders and content
- ✓ Securing CAS data access
- ✓ Scheduling regular backups
- ✓ Auditing
- ✓ Monitoring
- ✓ Troubleshooting

How You Perform Some Tasks Has Changed

Done the Kubernetes way

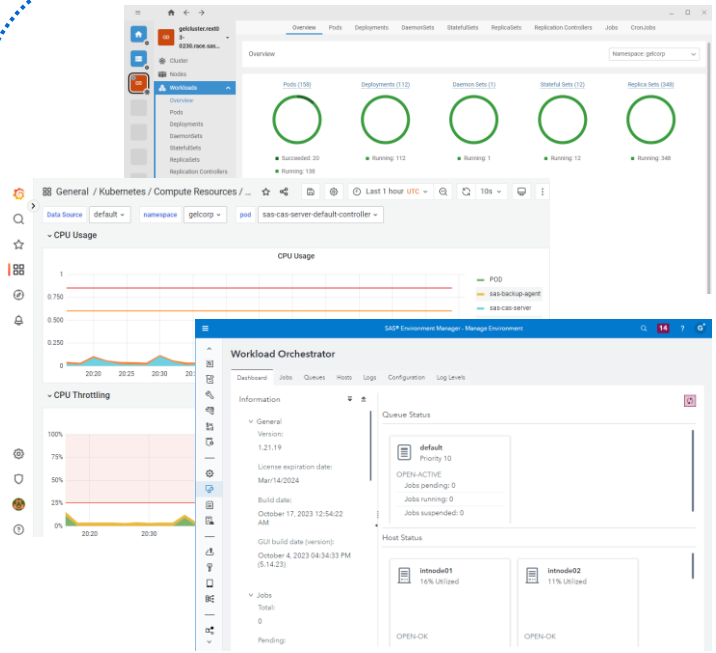
Remotely

Repeatable



Persistent

Variety of Administration Tools



Visual interfaces

$ kubectl

$ sas-viya

pyviyatools

REST APIs

Scripting

Command interfaces

Administration on Kubernetes

Common Work Patterns

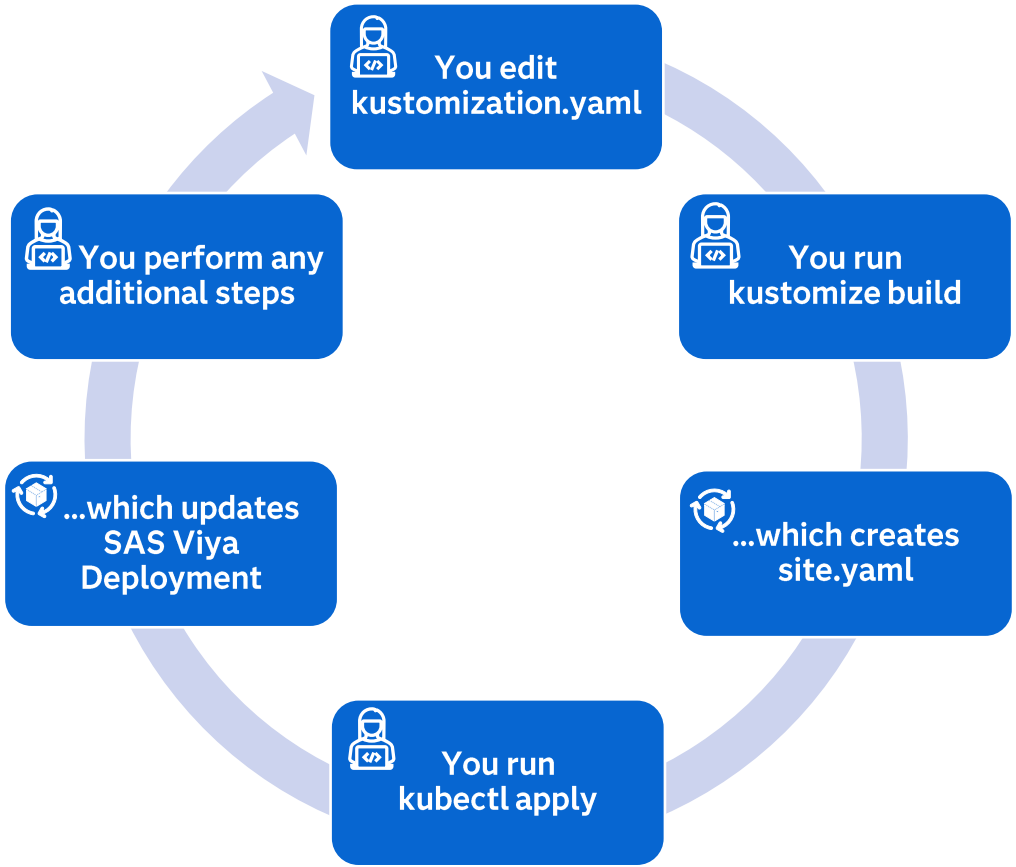
Making Configuration Changes

kustomization.yaml

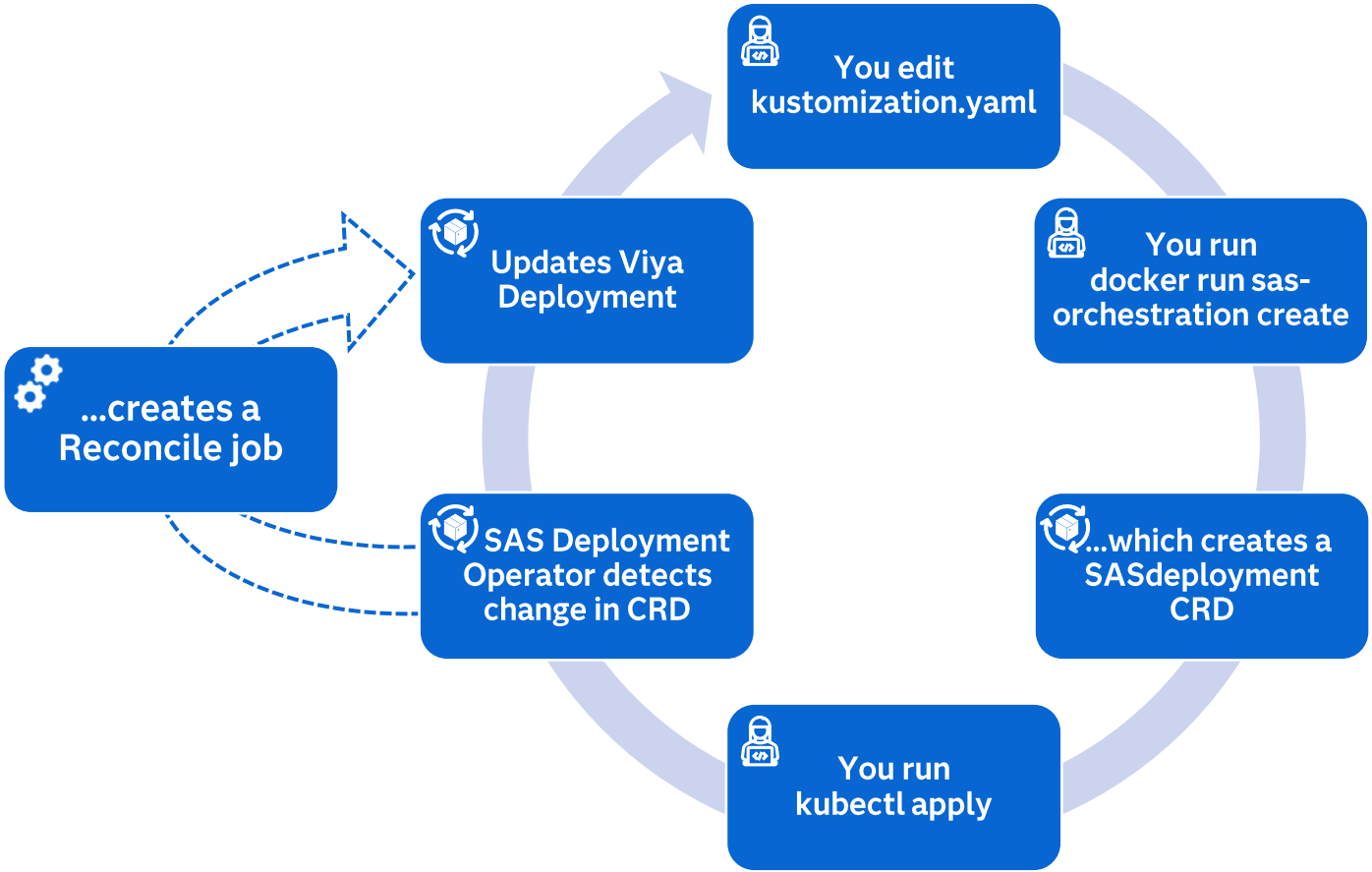
- SAS services and servers
- Persistent storage
- CAS servers
- TLS
- Image repository
- And more

```
namespace: gelcorp
resources:
  - sas-bases/base
  - sas-bases/overlays/network/networking.k8s.io
  - site-config/security/openssl-generated-ingress-certificate.yaml
  - sas-bases/overlays/cas-server
  - sas-bases/overlays/crunchydata/postgres-operator
  - sas-bases/overlays/postgres/platform-postgres
  - sas-bases/overlays/internal-elasticsearch
  - sas-bases/overlays/update-checker
  - site-config/sas-prepull/add-prepull-cr-crb.yaml
  - sas-bases/overlays/cas-server/state-transfer
  - sas-bases/overlays/sas-workload-orchestrator
  - site-config/sas-microanalytic-score/astores/resources.yaml
  - site-config/gelcontent_pvc.yaml
  - site-config/sas-open-source-config/python
configurations:
  - sas-bases/overlays/required/kustomizeconfig.yaml
transformers:
  - sas-bases/overlays/internal-elasticsearch/sysctl-transformer.yaml
  - sas-bases/overlays/startup/ordered-startup-transformer.yaml
  - site-config/cas-enable-host.yaml
  - site-config/sas-open-source-config/python/python-transformer.yaml
  - sas-bases/overlays/sas-programming-environment/watchdog
  - site-config/sas-programming-environment/lockdown/enable-lockdown-access-methods.yaml
  - sas-bases/overlays/required/transformers.yaml
  - site-config/mirror.yaml
  - sas-bases/overlays/internal-elasticsearch/internal-elasticsearch-transformer.yaml
  - sas-bases/overlays/sas-programming-environment/enable-admin-script-access.yaml
  - sas-bases/overlays/cas-server/state-transfer/support-state-transfer.yaml
  - site-config/change-check-interval.yaml
  - sas-bases/overlays/sas-microanalytic-score/astores/astores-transformer.yaml
  - site-config/sas-pyconfig/change-configuration.yaml
  - site-config/sas-pyconfig/change-limits.yaml
  - site-config/cas-add-nfs-mount.yaml
  - site-config/cas-add-allowlist-paths.yaml
  - site-config/compute-server-add-nfs-mount.yaml
  - site-config/cas-modify-user.yaml
components:
  - sas-bases/components/crunchydata/internal-platform-postgres
  - sas-bases/components/security/core/base/full-stack-tls
  - sas-bases/components/security/network/networking.k8s.io/ingress/nginx.ingress.kubernetes.io/ful
patches:
  - path: site-config/storageclass.yaml
```

Apply Configuration Changes with Kubernetes Commands



Apply Configuration Changes with Deployment Operator



Apply Configuration Changes with sas-orchestration deploy

Method	Benefits and Capabilities	Restrictions, additional considerations	Target use case
Deployment Operator	· **Automates deployment steps** (pull assets, kustomize, kubectl apply) · During updates, **automates** the pull of newer deployment assets and **several Deployment notes steps** · Manage the deployment / update of multiple Viya deployments of the same version within the same cluster · Feature to **schedule daily automated release updates** (within the same version) · **Validation of adherence to the Viya policy** (e.g.: Version update path controls)	· Requires **service account** to run the operator **with cluster-wide RBAC permissions** in the cluster · Requires **docker on the jump host** to generate the SAS Deployment CR · **No capability to apply resources based on permissions level** (cluster-wide, namespace-scoped, etc...) · Takes **longer to apply configuration changes**. For most (all ?) configuration changes - like changing the number of CAS workers - the reconcile operation redeploy, restart postgres instances. · **Increased complexity of configuring a private registry** for the Viya images. · Use in **offline/dark sites** (no internet access) **requires extra httpd server with a mirror of the entitlements**	· Operator driven Viya deployment happening within the cluster (it might meet the customer preferences or requirements). · SAS-Supplied automation ("Black box*" deployment).
sas-orchestration commands	· **Automates deployment steps** (pull assets, kustomize, kubectl apply) · During updates, automates the pull of newer deployment assets and **several Deployment notes steps**. · **Does not require a service account to run the operator** with cluster-wide RBAC · **Validation of adherence to the Viya policy** (e.g.: Version update path controls) ·The sas-orchestration image **includes a supported version of kubectl and Kustomize**	· Requires **docker on the jump host** to deploy SAS Viya · **No capability to apply resources based on permissions level** (cluster-wide, namespace-scoped, etc...) · Takes **longer to apply configuration changes**.	· Container-driven Viya deployment · A Docker image, "sas-orchestration", is used to automate the deployment process from outside the cluster. · SAS-Supplied automation.

Method	Benefits and Capabilities	Restrictions, additional considerations	Target use case
Kubectl apply commands ("manual")	· Better **control of the process** · Kustomize-generated **manifest file is available for troubleshooting and further processing** (for example security scans). · The most flexible method when integrating with customer automation tooling · **Declarative - The kubernetes way**	· **No SAS-supplied automation** features · **No validation of adherence to Viya Policy**	· Full control is required · Customers who want to use standard native Kubernetes deployment process (kubectl commands) · Customers with strict requirements on Kubernetes permissions with distinct roles to perform deployment and configuration operations.
SAS Viya 4 Deployment GitHub project	· **Automates everything** (initial config, pull assets, kustomize, kubectl apply) · **Automates several common customizations** documented in component's specific READMEs ·The **source code is open** and can be customized/changed if needed · Can use either the Deployment Operator Method or the sas-orchestration commands · Can **automate the pre-requisite installation** (ingress nginx, nfs provisioner, etc…) · Can **automate Viya monitoring tools deployments**	· Same **considerations as "Deployment Operator" or "sas-orchestration"** command methods · Specific **way to change the configuration, with adjustments** · **Does not perform updates** (documented steps must be followed with adjustments regarding the directory structure) · **Not supported for Red Hat OpenShift** · Issues with the Deployment method fell under the **sassoftware GitHub support policy (instead of standard support)**	· Simplified initial deployments, with Maximum automation (PoC, tests) · Combination/ Integration with the IaC · All target use cases listed for "Deployment operator" and "sas-orchestration commands" methods · SAS-provided open-source deployment automation

Requirements for User Accounts and Services

Problem: Running commands on specific Kubernetes resources?
Solution: Use labels!

Labels: Name-value pairs describing resources (environment, role, etc.).
Benefits: Efficiently target subsets of resources.
Ideal for commands as object names and count can change.
Useful for multi-pod components like SAS Viya services.

View labels using kubctl cmd: kubectl get <object-type> --show-labels
Select resources: Use labels in commands (e.g., kubectl delete pod -l app=my-app)

Example: Scale up CAS pods in "dev" environment:
kubectl scale deployment cas --replicas=3 -l environment=dev

Benefits for SAS Viya: Easily manage complex deployments like Postgres, CAS, and Consul.

Remember: Labels are powerful but don't define core system behavior.

Required role and additional permissions

- A recommendation of the official SAS Documentation is to provide "*a delegated, namespace-level administrator with the "get", "list," and "watch" permissions on all resources cluster-wide."*
- Ideally, the namespace administration privileges would be enough (without the need to give permission on all resources cluster-wide). What we can do is to use one of the Kubernetes existing "built-in" roles.

Required role and additional permissions

- Here is an example on how to give this role to a “viya-admin-lab” service account.

```
# create the manifest
  tee /tmp/create-binding-for-viya-admin-lab.yaml > /dev/null << EOF
  ---
  apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
  metadata:
    name: sas-rolebinding-namespace
    namespace: lab
  subjects:
  - kind: ServiceAccount
    name: viya-admin-lab
    namespace: lab
  roleRef:
    kind: ClusterRole
    name: cluster-admin
    apiGroup: rbac.authorization.k8s.io
  EOF

# apply the manifest
  kubectl apply -f /tmp/create-binding-for-viya-admin-lab.yaml
```

- We simply create a “RoleBinding” object to attach the built-in “cluster-admin” ClusterRole to our “lab” namespace. So, while it is a “ClusterRole”, it can be limited to the scope of a particular namespace.

- With the above privilege definition, someone using our “viya-admin-lab” service account could change everything inside our Viya namespace but could not do anything else outside of the namespace.
- Like the other 99% of the “cluster-local” label *

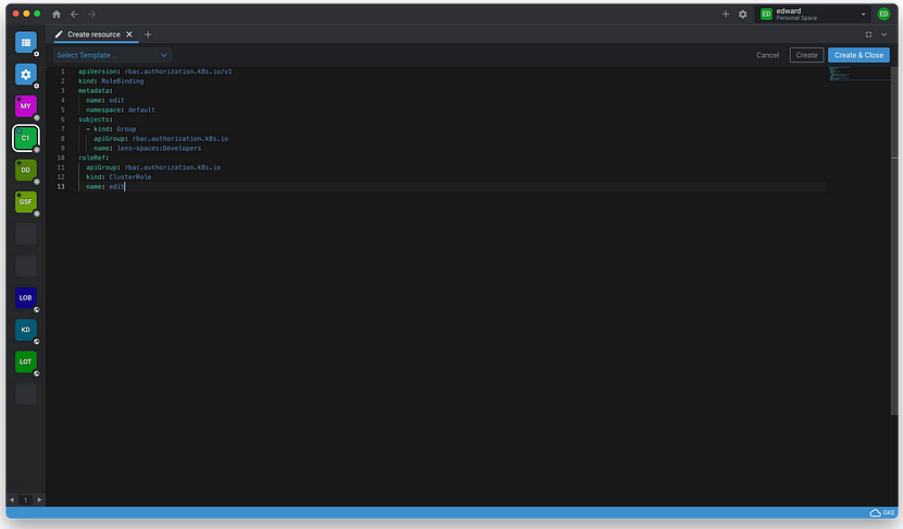
* Need to create the “viya-admin-lab” service account and an associated kubeconfig for this user.

Break down the responsibilities

- The breakdown of tasks between the Kubernetes administrator and our SAS Viya administrator using the service account described above
  - with privileges limited to the SAS Viya namespace.

Task	Responsibility	Notes
Deploy SAS Viya	Shared	The Kubernetes administrator runs the "kubectl apply" commands for "cluster-api", "cluster-wide" and "cluster-local" levels. The SAS Viya administrator runs "kubectl apply" command for the namespace level (and potentially for the "cluster-local" level).
Update SAS Viya	Shared	
Stop/Start the SAS Viya platform	SAS Viya administrator (namespace admin)	This task is handled by Cronjobs defined at the namespace level.
Backup and restore the SAS Viya environment	SAS Viya administrator (namespace admin)	
Perform common customizations (change CAS or SAS programming runtime settings, change CAS topology, etc...)	SAS Viya administrator (namespace admin)	While the SAS Viya administrator cannot change the Custom Resource Definitions (CASDeployment definition), he can modify Custom Resource instances (CASDeployment instances), which is enough to change the configuration of a deployed CAS Server.
Allocate additional permissions to the SAS Viya administrators when/if needed for specific configuration.	Kubernetes Administrator	We haven't tested all possible configuration/customization. Some may exceptionally require additional privileges. The additional permissions would be added as required.

Configuring Kubernetes Role-Based Access Control with Lens Spaces



- Administrators typically do not want to expose their Kubernetes clusters on a public network directly
- either stay on a private network or create VPN access for remote work
- Lens Spaces, a feature within [Lens — The Kubernetes Platform],
- Kubernetes users can easily share access to a cluster securely. Lens Spaces utilizes end-to-end encryption to secure connections between users and clusters, eliminating the need for a VPN.

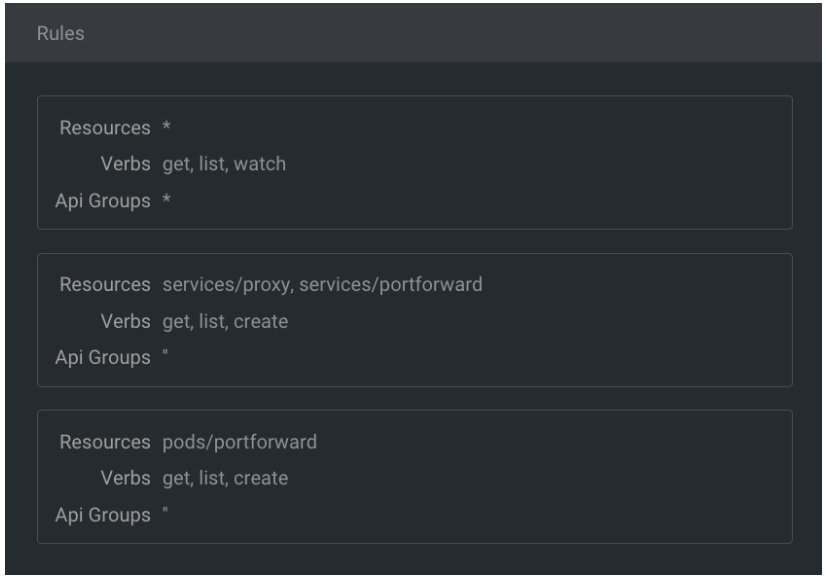
Configuring Kubernetes Role-Based Access Control with Lens Spaces

Lens Spaces built-in teams: **Members**, **Admins**, and **Owners**.

- **Members** — Have **Read Access** to all connected clusters in a space
- **Admins** — Have **Read & Write Access** to all connected clusters in a space
- **Owners** — Have **Read & Write Access** to all connected clusters in a space

To support these roles, all clusters connected to Lens Spaces are preconfigured with the following **Cluster Roles** and **Cluster Role Bindings**:

ClusterRole: lens-spaces-view



Summary

- Administration involves Kubernetes admins as well as SAS Viya admins
- Core SAS Viya administration tasks are still necessary
- Administrators will use a variety of tools to manage SAS Viya
- Many administration tasks fall into common patterns of work
  - Making configuration changes
  - Searching for and acting upon specific resources